Strategic Support of Algebraic Expression Writing

Mary A. Mark and Kenneth R. Koedinger

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract. The examination of user data as a basis for developing production models of user behavior has been a major focus in the PAT Algebra I Tutor's development. In recent work, we have investigated relationships between related tasks and the solution strategies displayed by students. To solve a PAT Algebra I problem, students must complete several related arithmetic and algebraic tasks. The sequences in which these tasks are completed suggest problem-solving strategies of students. We have observed a characteristic pattern of students' success rates on related tasks. We have also observed that students' success on specific skills (e.g. constructing a symbolic representation) may differ depending on whether students previously carried out related tasks in the same problem (e.g. solving an analogous arithmetic question). This information has important implications for our user model and our modeling approach.

1 Introduction

Users with different experience and expertise may interact quite differently with the same application. As users move up the not-so-gentle slope from novice to expert, the strategies they use often change (Siegler and Jenkins, 1989). Ideally, user modeling applications should be able to track the use of multiple strategies by users, and suggest desirable strategies for individual users.

One ubiquitous pattern of change in the process of skill acquisition is the transition from more concrete instance-based modes of interaction (e.g. direct manipulation in spreadsheets or word processors) to the use of more abstract symbols and functions (e.g. use of formulas and style specifications). In mathematics, the concrete-instance to abstract-symbol transition corresponds broadly to the advance from arithmetic to algebraic competence. We are investigating this pattern of development with an intelligent tutoring system for grade 9 algebra, the PAT Algebra I Tutor.

In traditional views of problem-solving in this domain, students must first develop equation solving competence and then use equations to solve story problems. Even research into multiple solution strategies has focused on symbolic manipulations (Mayer, 1982). However, cognitive research by Koedinger and Nathan (1999) shows that initially students can comprehend and solve quantitative constraints better when constraints are presented in concrete verbal and numerical form rather than abstract symbolic form. Students use concrete instance-based strategies to solve quantitative constraints, in addition to the formal translate-to-algebra strategy. Instruction which connects concrete strategies to abstract ones is more effective (Koedinger and Anderson, 1998).

In this paper we present a detailed analysis of on-line data collected by PAT's user model. We focus on a set of skills related to the concrete-instance to abstract-symbol transition in mathematics, involving solution of concrete cases and writing of abstract expressions. The data collected enables us to identify different strategies chosen by students, the proportion of students choosing them, and student success rates at the tasks involved. One concern is to compare the relative difficulty of these skills. We also want to know if solving one task in a problem enables students to more successfully complete related tasks in the same problem. Is writing an expression after successful completion of a related skill really the same task as writing an expression without previously completing a related skill? How should this be modeled? Also, at a strategic level, do students prefer certain solution paths? If working on concrete cases can help students to write expressions, are students making strategic choices that reflect this? To what extent can our existing tutors track and support more effective strategy use by students? What implications does this have for our approach to student modeling?

2 The PAT Algebra I Tutor

The PAT Algebra I tutor focuses on the mathematical analysis of real world situations and the use of multiple representations for problem-solving. Checking charges on a phone bill and comparing the costs of different car rentals are examples of such real-world situations. To solve a PAT problem, students read a textual description of a situation with some questions. Students use multiple representations of the situation including words, numbers, symbols, tables, and graphs, to reason about the situation and answer questions. (Koedinger et al., 1997). At the tutor's core is a production rule model of student behavior which enables the tutor to diagnose correct and incorrect actions, provide help and feedback, and control the student's progress through the curriculum. Previous papers discuss the overall educational impact of PAT (Koedinger et al., 1997; Koedinger and Sueker, 1996) and learning of expression writing skills (Mark et al., 1998).

Figure 1 shows a partial solution for a single linear equation problem from Lesson 1 of the PAT Algebra I curriculum for the 1996-1997 school year. In Lesson 1 the student must construct a table by finding solutions to questions and by writing an expression. More complex problems and additional tools are introduced in later lessons. The Problem Description (upper left of Figure 1) describes the cost of skating at Sky Rink, based on a flat fee for renting skates and a per-hour fee for time at the rink. Students investigate the situation using a spreadsheet tool (the "Worksheet" window) and a specialized tool for identifying algebraic expressions from concrete cases (the "Pattern Finder" window). Students construct the Worksheet (lower left) by labeling the columns with quantities from the situation, entering units, writing an algebraic expression, identifying givens, and answering the numbered result unknown questions. Once the relevant quantities have been identified a student can complete the rest of the table in any order desired. The Pattern Finder (center right) can be used at any time to work out the form of the algebraic expression. On average, students spend about twelve minutes solving a tutor problem like this.

The tutor monitors and responds to students through the use of a production rule based user model. As detailed in previous papers (Koedinger et al., 1997; Mark et al., 1998), *model tracing* monitors student behavior within a problem, diagnosing student success and failure on individual skills, and generating help and feedback, through the matching and firing of production rules. The "Messages" Window (lower right of Figure 2) displays these messages to the student. *Knowledge tracing* monitors learning across problems. Information about the success and failure of a student's attempts at a skill is incorporated through mathematical modeling into a current assessment of the likelihood that the student has learned the skill. Assessments of what the student knows are used to individualize the tutor's curriculum, assigning remedial problems in areas where the student is weak. The tutor's current assessment of the student's learning is shown in the skillometer window (upper right: entitled with the student's name, e.g. "Mary Mark".)

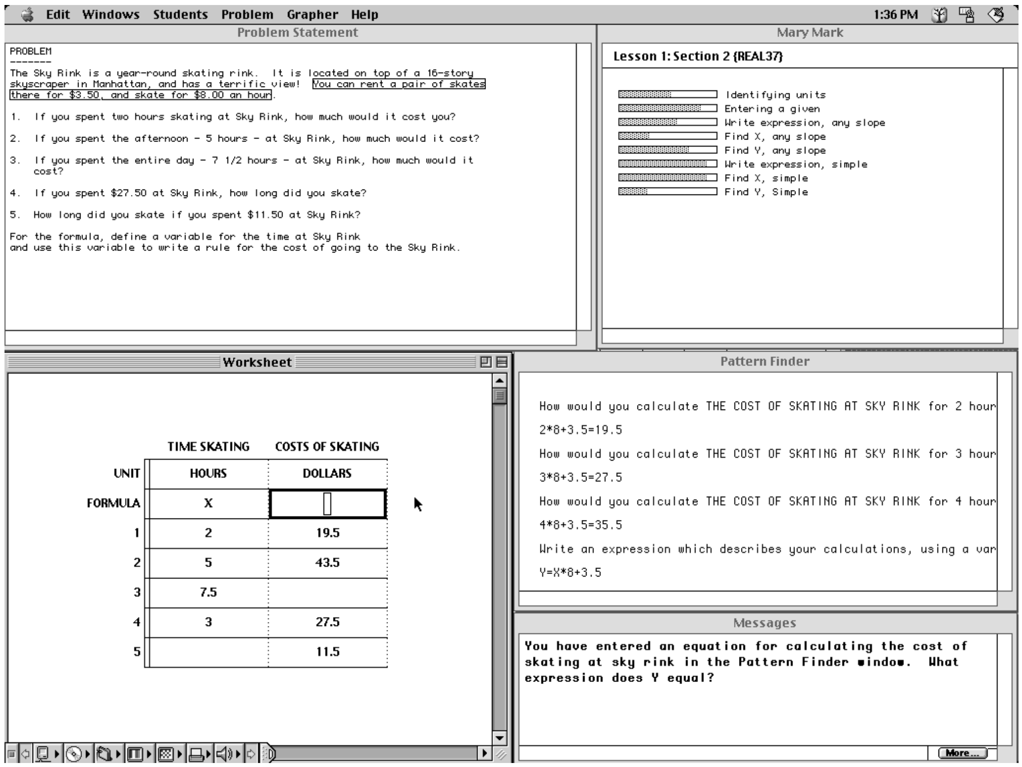

Figure 1. The PAT Algebra I Tutor, Lesson 1.

3 Expression Writing and Related Skills

One of the primary skills that the student is expected to learn is the writing of an algebraic expression to describe a problem situation. The ability to capture complex mathematical relationships in a concise algebraic form is seen as a meaningful indicator of students' understanding of a situation and the mathematical relationships involved. The ability to translate a quantitative problem situation into algebraic symbols and to use that symbolic form is important for effective use of real-world tools like symbolic calculators (Koedinger and Anderson, 1998).

Writing an expression requires the composition of an algebraic representation involving a variable, numbers, and operators. To experts, writing an algebraic expression like "8.0*x+3.50" to describe the Sky Rink problem seems like a fairly obvious process: you read the problem statement, identify the changing quantity and rate of change in the problem, check for an initial value, and put the appropriate numbers into the appropriate places in the formula. Novices find it considerably more difficult, and are affected by features of the problem situation such as the use of integer vs. non-integer numbers, and the presence and sign of y-intercepts (Mark et al., 1998).

Novices display more success with the type of concrete case called a *Result Unknown* (Koedinger and Anderson, 1998). Question 2 in the "Problem Statement" window involves a result unknown: "If you spent two hours skating at Sky Rink, how much would it cost you?" The student is given a numeric value for the number of hours skated and asked to find the resulting value for the cost. Correct answers include the final calculated value, (e.g. "19.50") and any arithmetic expressions evaluating to this answer (e.g. "3.5+16" or "8+8+3.5"). Students may solve result unknowns by translating the problem into symbolic form and then substituting a value for the variable. They may also engage in a process of arithmetic problem-solving in which the problem is broken down into arithmetically tractable sub-tasks which are completed to obtain a result without ever combining the sub-tasks in ways that symbolically relate them. For example, a student calculating the cost of skating at Sky Rink for 2 hours might multiply 2 times 8 first, and then take this intermediate result and add 3.5 to get 19.5. These two operations are usually performed as separate arithmetic steps with equal signs in a calculator, or as separate summations in column arithmetic on paper. When asked how they obtained their answer, students will often respond "I added 16 and 3.5" and have trouble remembering how they got 16. This strategy avoids the application of skills for *composing* the two operators into a single symbolic sentence. Work by Heffernan and Koedinger (1997) suggests that composing an arithmetic symbolization may be almost as hard for students as writing an expression with a variable.

The "Pattern Finder" isolates this composition skill by asking students to write a concrete symbolization, a *Pattern Instance*. In the first concrete case in the "Pattern Finder" window, students are asked "How would you calculate the cost of skating at Sky Rink for 2 hours?" The student must indicate how he or she would get an answer, by entering a symbolic mathematical expression as a solution for the cost (e.g. "2*8+3.5"). The numeric result (e.g. "19.5") or a random mathematical expression which yields it (e.g. "10+9.5") will not be accepted. When a correct answer is given by the student (e.g. "2*8+3.5"), the Pattern Finder displays the calculation and its result as an equation (e.g. "2*8+3.5=19.5").

The Pattern Finder is structured so that students work through a progression of pattern instances, using a fixed sequence of values for x of 2, 3, and 4. After the student has solved three or more such concrete cases, he is requested to "Write an expression which describes your calculations, using a variable." In this final *Generalization* step the student is expected to examine the preceding arithmetic symbolizations for underlying patterns, and make the generalization to using a variable in place of the changing numeric value.

Consider which of these the steps in the Pattern Finder window is likely to be the most difficult for students. Is it writing an expression when x=2, x=3, x=4, or explicitly using the variable x? Most people, when asked this question, predict that the final step of generalizing from the arithmetic expressions to the algebraic expression is the most difficult. Indeed, this was our original intuition when designing the Pattern Finder. We hoped that doing multiple instances (2, 3, 4) would help students make this difficult generalization step. Data collected by PAT's user model is particularly interesting with respect to this question!

4 Student Success Rates on Related Skills

Data was collected by the PAT tutoring system during the 1996-1997 school year at Langley, a typical urban high school. Classes at Langley used the PAT tutor as part of their regular grade nine algebra I program, spending two out of five 40-minute classes per week on the tutor. As students worked, the PAT tutor saved protocol files recording student actions, their success or failure, and the production rules fired. The production model of the 1996-1997 tutor did not identify skills by order of completion, so we generated the necessary sequence-based production information from the protocol data during analysis. The data presented comes from 75 students, who completed a total of 1026 problems in the first lesson of the grade nine tutor curriculum (an average of 13.68 per student). Lesson 1 contained a demonstration problem (ignored in this analysis), 8 required problems, and a pool of 16 remedial problems. Students worked independently through the lesson. The basic curriculum was individualized for each student as the tutor assigned differing orders and numbers of problems. Not all students completed the lesson.

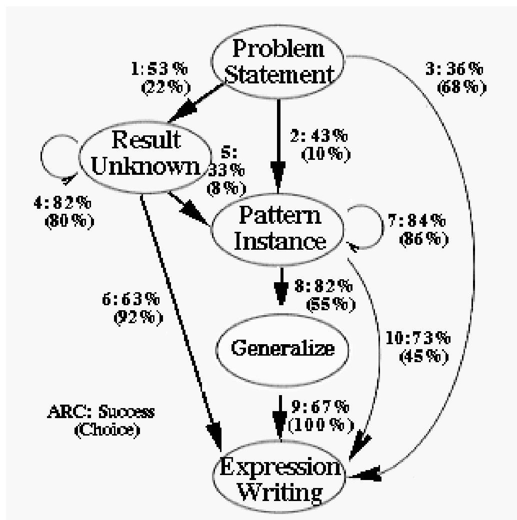

Figure 2. Student solution paths.

Figure 2 shows a partial graph of the sequences in which a student can complete the related skills when working through a problem. A node in the graph represents a state in which certain tasks have been successfully completed. A student may begin the problem by writing the expression, or do one or more result unknowns first, or use the Pattern Finder. In the Pattern Finder, a student must complete at least three pattern instances before the generalizing step. It is also possible to change course mid-way and skip from the Pattern Finder to writing the expression in the Worksheet, for example. Unrelated skills such as identifying quantities and units and entering givens were ignored. Since our main concern was to examine the effect of concrete cases on writing of expressions, we also ignored activities which occurred after writing Worksheet expressions.

For any transition in the graph, we can calculate the percentage of students in a state who follow a particular solution path out of that state by successfully completing a related action. We can also calculate the percentage of students who make this transition successfully on their first attempt at the transition skill. For the purposes of this analysis, to ensure that the success rates are comparable, succeeding at one of the related skills causes the other related skills to be treated as if they have not been previously attempted. Thus, success rates at any transition reflect the most immediate state of the student's knowledge and abilities. In this way, we avoid the problem of contamination due to a change in strategy: unsuccessfully attempting a skill early on will not affect the success rate calculated for that skill after completing a sequence of other related tasks. Table 1 contains a summary of these skill transitions, the cognitive task components which we believe are involved, and the strategy choice and success rates observed.

Table 1. Success rates and strategy choices.

Sequence of Skills Completed	Cognitive Task Components	Success Rate	Strategy Choice
1. Problem -> Result Unknown	Concrete case: Symbolization or Arithmetic problem solving	53%	22%
2. Problem -> 1st Pattern Instance	Concrete case: Symbolization	43%	10%
3. Problem -> Expression	Symbolization and Generalization	36%	68%
4. Result unknown -> Result unknown	Concrete case: Symbolization or Arithmetic problem solving	82%	80%
5. Result unknown -> Pattern	Concrete case: Symbolization	33%	8%
6. Result unknown -> Expression	Symbolization and Generalization	63%	92%
7. Pattern -> Pattern	Concrete case: Symbolization	84%	86%
8. Pattern -> Generalization	Generalization	82%	55%
9. Generalization -> Expression	Transfer	67%	100%
10. Pattern -> Expression	Generalization	73%	45%

The students' first actions give us a comparison of the relative difficulty of writing an expression, a pattern instance, and a result unknown. Students succeeded 36% of the time on their first attempt to write an expression, 43% on their first attempt at a pattern instance, and 53% on their first attempt at a result unknown. This supports our hypothesis that abstract symbolizations involving variables (expression writing) are harder for students than the solution of concrete cases (result unknown, pattern instance). This replicates the pre-and-post-test results for symbolization and result unknown skills of Koedinger and Anderson (1998), at the level of specific traceable skills. Students also found it harder to write their first pattern instance (a concrete case requiring symbolization) than to solve their first result unknown (a concrete case not requiring symbolization). This agrees with the composition effect reported by Heffernan and Koedinger (1997), in which solving the whole was more difficult than finding the sum of its parts, independent of variable use. However, PAT students found it easier to write a pattern instance than an expression when using the tutor. Somewhat in contrast, Heffernan and Koedinger reported only a small not significant effect of variable (abstract symbolization) vs. no variable (pattern instance) problems.

If we follow students through a sequence of tasks in the Pattern Finder (center right of Figure 1), we can see that students improve substantially from their first pattern instance (43%), to subsequent pattern instances (84%). Breaking this out into the first, second, third, etc. pattern instances showed a substantial increase after the first pattern was solved, and a much smaller increase for each successive instance. This supports the hypothesis that completing the arithmetic symbolization in the first concrete case is a important step enabling students to more effectively solve further pattern instances. However, the transition from the concrete cases to the abstract symbolization (82% success) is only marginally harder than doing another concrete case (84%). In contrast to our expectation, students did not have great difficulty making this generalization step (82%); rather, it was writing the first pattern instance (43%) that was most difficult for them.

Comparing the first result unknown in a sequence with later result unknowns we see that students have the most difficulty solving their first result unknown (53%), and improve considerably on later result unknowns (82%). Again, the first concrete case was the most crucial, with small increments in success rate for each later concrete case. However, after writing a result unknown, it is still substantially harder to write an expression (63%) than to do another result unknown (82%). This supports the idea that students who solve result unknowns may be doing so either by engaging in a concrete case symbolization, or through a process of arithmetic problem solving which does not involve symbolization, while students who write pattern instances are required to compose symbolic representations.

Students are more successful when writing an expression after completing some result unknowns (63%) than when writing an expression without completing any concrete cases (36%). Students using the Pattern Finder were more successful at writing expressions in the Worksheet after completing only pattern instances (73%) or both pattern instances and the generalization step (67%). While these two success rates are similar, the types of errors displayed on the transitions are different. Students who completed the generalization in the Pattern Finder had more difficulty transferring the expression back to the Worksheet than we expected. However, they did tend to make errors consistent with a transfer attempt (e.g. "Y=X*8+3.5" for "X*8+3.5"). The errors of students who went from a pattern instance to writing a Worksheet expression suggest serious difficulties in symbolization. Such students are 20% more likely to enter unrelated or unparseable solutions, and 22% more likely to request help. Protocols suggest that we may have two populations of students on this arc: one which is somewhat skillful, and one which has great difficulties.

5 Solution Paths and Strategic Choices

An examination of solution paths shows that 68% of the time, students successfully wrote the expression before completing any concrete instances in the Pattern Finder or the Worksheet. This preference may partly reflect the layout of the Worksheet: the expression appeared at the top as in a spreadsheet. Twenty-two percent of the time, students successfully calculated a result unknown first. Only 10% of the time did students begin by completing the first pattern instance in the Pattern Finder. Eighty percent of the time, students who successfully calculated one result unknown did additional result unknowns before writing the expression. After one or more result unknowns, 92% of students wrote the expression, while 8% did further concrete cases with the Pattern Finder. Eighty-six percent of the time, students who completed one pattern instance did additional concrete cases with the Pattern Finder before generalizing in the Pattern Finder or writing an expression in the Worksheet. Fifty-five percent of students who completed at least one pattern instance went on to generalize an expression in the Pattern Finder, while 45% went from the concrete instances to writing the expression in the Worksheet. Although there is evidence to suggest that students can benefit by doing concrete instances before writing expressions, students do not generally use the Pattern Finder for this. More often they solve result unknowns, but in general they follow the layout of the table and immediately try to write an expression.

A student may not depend on a single strategy while using the tutor. "Student A" is a particularly good case study for demonstrating that a student may acquire strategic knowledge about what solution paths to follow, as well as learning individual skills. "Student A" illustrates several solution strategies which students may use when working on tutor problems. Her early solution paths are somewhat erratic, but she starts to use the Pattern Finder consistently as a support tool, and then, as she becomes more proficient, leaves it behind.

In Lesson 1 the teacher introduced the tutor by working through a demonstration problem in a recommended sequence, identifying quantities and units in the Worksheet, and then completing the Pattern Finder. The Pattern Finder was recommended as a tool which students could use to find expressions if they were having difficulty. "Student A" follows the teacher through the MX demonstration problem without difficulty. She then solves her first randomly assigned problem, following the suggested sequence of steps, and starts a third problem before the class adjourns.

Despite this promising beginning, she has considerable difficulty in subsequent classes. Her initial answers are often correct or nearly correct but she frequently enters them in inappropriate cells, putting an expression in place of a variable or switching a given and a result. She is easily distracted by additional information in the problem (other givens or distractors). Once she gets off-track, she tends to flounder, and to generate increasingly unlikely solutions.

In her fourth problem, "Student A" enters a sequence of pattern instances and then, without entering an abstraction, goes directly to entering an expression in the Worksheet. She gets it wrong because she puts the expression in the wrong column. Several similar error sequences seem to convince her that it is better to work through a complete Pattern Finder sequence before entering the expression in the Worksheet.

In problems 8-12, "Student A" consistently goes to the Pattern Finder before attempting the expression or any of the concrete cases in the table. She succeeds about 50% of the time in her first attempt at a pattern instance, but once she has completed one (e.g. "8.0*2+3.5"), she almost invariably writes the rest without difficulty (90% success), and generalizes correctly (100% success). She is increasingly successful at transferring the expression back to the Worksheet.

In problems 11 and 12, "Student A" is able to write the first pattern instance without errors, even though she sees a problem of the form MX-B for the first time in Problem 12. These successes appear to give her confidence, and in problems 13 and 14, she labels the Worksheet columns, enters units, indicates the variable (X) and correctly enters the MX expression, without using the Pattern Finder. Her next problem is an MX+B form. Unfortunately she overgeneralizes from the previous two problems, and her initial MX solution is incorrect.

In many ways, "Student A" displays an ideal pattern of strategy choice. When in difficulties, she turns to the Pattern Finder as a useful tool, and uses it until she considers herself to be mastering the relevant skills, at which point she discards it. In the same way, an expert tutor might suggest that a student use tools for support on an as-needed basis.

Other students find the result unknowns in the table useful as concrete cases. "Student B" shows a consistent pattern of strategy choice (19/24 problems) in which he solves result unknowns in the Worksheet before entering an expression. In other cases (4/24), he asks for help for a result unknown and then attempts the expression directly, without completing a result unknown. His success rates show a characteristic pattern: he is successful on 30% of his first result unknowns, but his success rate on subsequent result unknowns improves to 70%. His success rate for entering a subsequent expression, however, is only 60%, indicating that symbolization with a variable still gives him difficulty. When he attempts an expression without first completing concrete cases, his success rate is only 50%.

Students who tend to go straight for the expression without doing concrete cases vary considerably in skill. Students "C" and "D" consistently complete the expression before doing other work, and succeed on 64% and 80% of their first attempts at expression writing, respectively. In contrast, Students "E" and "F" succeed about 24% and 30% of the time. These low-achieving students could potentially benefit from working out concrete cases first, but tend not to do so.

6 Conclusions

Deriving expressions through induction. We hypothesized that the cognitive tasks of symbolic composition and generalization of variables would affect students' success at expression writing and other related skills. Our results show that students find it easier to solve concrete cases that do not require symbolization than concrete cases that do; and that either type of concrete case is easier than writing an expression with a variable. Our results support the idea that students learn to construct expressions through induction from concrete cases. Requiring students to show their work symbolically can help them to make this transition. We found further support for the surprising result that students find composition much more difficult than generalization.

Several changes were made to the 1998-1999 PAT tutor and curriculum as a result of these observations. We reduced the number of concrete cases in each problem, since results suggest that solving the first concrete case in either the Pattern Finder or the Worksheet has the most impact, and subsequent cases yield little further improvement. We also moved the expression writing row to the bottom of the Worksheet after the result unknowns. This may prompt students to solve result unknowns before writing the expression. More substantial changes can also suggested:

Refining production models to reflect cognition. Our comparison of solution paths and success rates leads us to the conclusion that expression writing is not always the same skill. Depending on the sequence in which tasks are completed, tasks like expression writing may involve various cognitive components. PAT's productions should be redesigned to model both surface goals that students are trying to satisfy (expression writing, solving result unknowns, and writing pattern instances), and deeper underlying cognitive skills (symbolization, generalization, and transfer). However, a behavior like solving result unknowns, which is ambiguously related to underlying skills, may considerably complicate the tracing and attribution of production skills. Luckily, other researchers are already dealing with some of the issues this may involve, such as uncertain knowledge (Katz et al, 1994), and relating subskills and factors (VanLehn et al, 1998).

Identifying strategic behavior. As case studies for "Student A" and "Student B" suggest, students may strategically select solution paths when working on a problem. Students can learn strategic knowledge about approaching an overall problem, just as they learn individual skills. Many students, however, remain unaware of the possible benefits of strategic choices. Currently, the tutor allows students to follow a variety of strategies, but does not attempt to track their use of such strategies. Diagnosis of strategic choices along solution paths within problems could be achieved by writing production rules which are more sensitive to information about the student's current working context and solution state. We can diagnose strategic behavior within problems in the same way that we diagnose a student's current actions, in our existing modeling paradigm.

Reflective modeling of strategic behavior. A student makes strategic choices about how to solve a problem in part on the basis of self-assessments of his or her skills, as shown by "Student A" and "Student B". In our current modeling approach, information about the problem structure, the current solution state, and the student's behavior, is available to model tracing for diagnosis. Information that captures student behavior across problems is not accessible to model tracing. The tutor's assessments of what the student knows are available only to the knowledge tracing component which individualizes the curriculum. To determine not only what strategy choice a student is making, but what strategy choices a student should make, and when to recommend such actions, further information about the student is needed. Assessments of student knowledge are an important source of information, available in the tutor, which could be accessed by the production model. A reflective model that used assessments of a student's learning of skills could provide students with individualized strategic advice about desirable solution processes. "Student A", who experiences success with pattern instances, but not with result unknowns, could be given different help from "Student B", who is successful at writing result unknowns. Such a tutor could also distinguish between "Student D", who displays considerable skill at expression writing, and "Student E", who is unsuccessful at this skill. Developing a more self-reflective user model, which could utilize knowledge tracing information about individual students to teach strategic skills, would be an exciting extension of our current user modeling approach.

References

Heffernan, N. T., and Koedinger, K. R. (1997). The composition effect in symbolizing: The role of symbol production vs. text comprehension, *Proceedings of the Nineteenth Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Erlbaum.

Katz, S., Lesgold, A., Eggan, G., and Gordin, M. (1994). Modeling the student in Sherlock II. In Greer, J. E., and McCalla, G. I., eds., *Student Modelling: The Key to Individualized Knowledge-Based Instruction,* New York: Springer-Verlag. 99-125.

Koedinger, K. R., and Anderson, J. R. (1998). Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. *Interactive Learning Environments, 5,* 161-179.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., and Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8,* 30-43.

Koedinger, K. R. and Nathan, M. J. *The real story behind story problems: Effects of representations on quantitative reasoning.* Manuscript submitted for publication.

Koedinger, K. R. and Sueker, E. L. F. (1996). PAT goes to college. In *Proceedings of the Second International Conference on the Learning Sciences*, Charlottesville, VA: Association for the Advancement of Computing in Education.

Mark, M. A., Koedinger, K. R., and Hadley, W. H. (1998) Elaborating models of algebraic expression writing. In *Intelligent Tutoring Systems: 4th International Conference Proceedings (ITS'98)*. New York: Springer-Verlag.

Mayer, R. E. (1982). Different problem-solving strategies for algebra word and equation problems. *Journal of Experimental Psychology: Learning, Memory and Cognition, 8,* 448-462.

Siegler, R. S. and Jenkins, E. (1989). *How Children Discover New Strategies.* Hillsdale, N. J.: Lawrence Erlbaum Associates.

Van Lehn, K., Niu, Z., Siler, S., and Gertner, A. S. (1998) Student modeling from conventional test data: A Bayesian approach without priors. In *Intelligent Tutoring Systems: 4th International Conference Proceedings (ITS'98).* New York: Springer-Verlag.